UNIVERSITÄT
LEIPZIG

Optimizing Perceived Aesthetics of Mobile UIs Using Metric Guided Generative Pipelines

June 24, 2024

Moritz Wörmann

Supervised by Patrick Ebel & Niklas Deckers at
Junior Research Group CIAO (Computational Interaction and Mobility)
ScaDS.AI, Universität Leipzig

Problem Setting:
Creating Aesthetically Pleasing UIs

Background

- Interaction with software is predominately performed using Mobile Applications (Apps)

Background

- Interaction with software is predominately performed using Mobile Applications (Apps)
- The design is specific to the usecase and differs from other apps in order to stick out

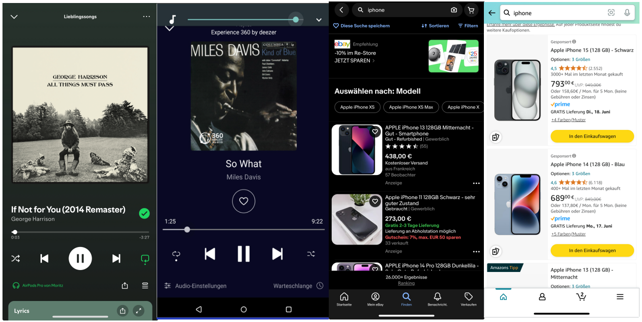


Figure 1: Different mobile applications

Aesthetics Are Key to Success

- Whether a UI is considered aesthetically pleasing is a key indicator for user satisfaction [1]

Aesthetics Are Key to Success

- Whether a UI is considered aesthetically pleasing is a key indicator for user satisfaction [1]
- Good products may still be considered bad if the corresponding UI is ugly

Current State of the “Art”

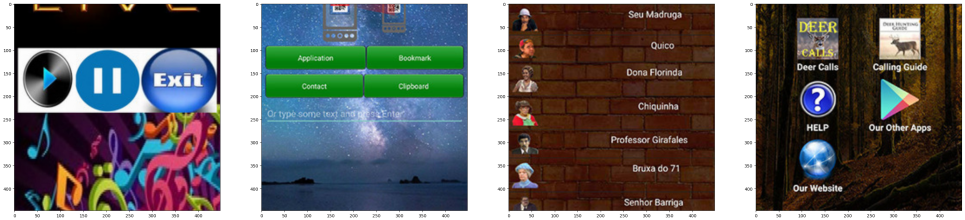


Figure 2: Reproduced from de Souza Lima et al. [2]

– Tools like AppInventor lead users to create unaesthetic designs

Design Process of Mobile UIs

Creating complex User Interfaces can be a lengthy process:

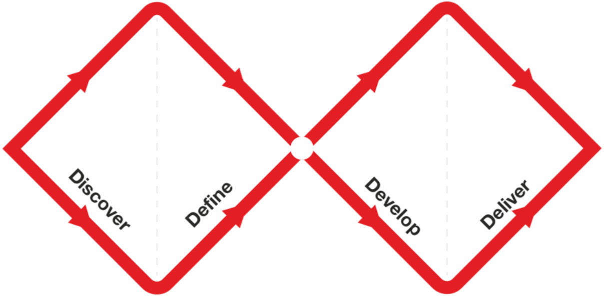


Figure 3: The Double Diamond Model, reproduced from Design Council [3]

– Usability and UX of UI is determined by functionality and aesthetics

What Even Is Considered Pretty?

- Without knowing the (potential) userbase, designers don’t know what their customers will consider pretty.

What Even Is Considered Pretty?

- Without knowing the (potential) userbase, designers don’t know what their customers will consider pretty.
- User studies are the main approach to assess preferences

What Even Is Considered Pretty?

- Without knowing the (potential) userbase, designers don’t know what their customers will consider pretty.
- User studies are the main approach to assess preferences
- Users may not agree on what is considered pretty [4]
- User studies beyond scope for developers

→ Reuse existing datasets and models for determining aesthetics of given UIs

Problem Setting

- Identified arrangement of elements as key factor contributing to perceived aesthetic of UIs [5]
- Given a rudimentary UI layout with functional elements

Problem Setting

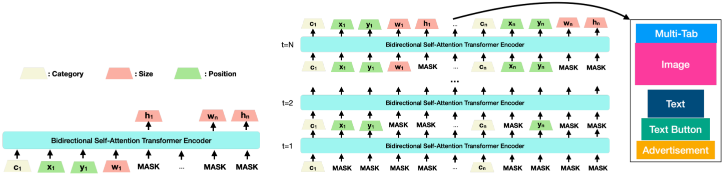
- Identified arrangement of elements as key factor contributing to perceived aesthetic of UIs [5]
- Given a rudimentary UI layout with functional elements
- Arrange UI elements in aesthetic way automatically without disrupting functionality

Related Work

Related Work

- Both algorithms focus on (1) generation from scratch and (2) generation based on predefined elements

Transformers for Layout generation: BLT



(a) BLT Training Phrase.

(b) BLT Iterative Decoding Process.

Figure 4: Reproduced from Kong et al. [6]

Automated Layout Generation: LayoutDM

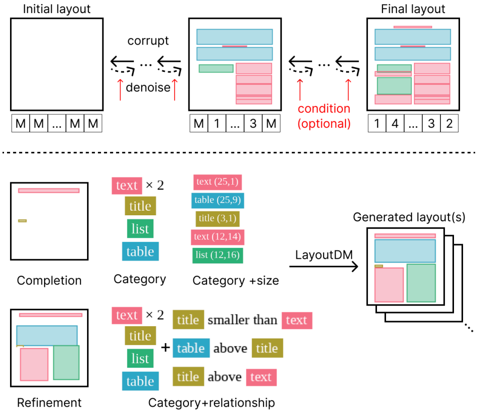
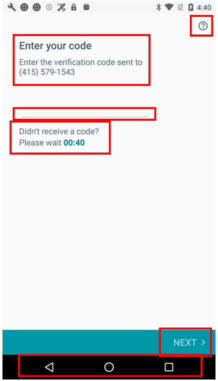


Figure 5: Reproduced from de Souza Lima et al. [2]

- Related work focuses on layout generation without being guided by metrics like aesthetics

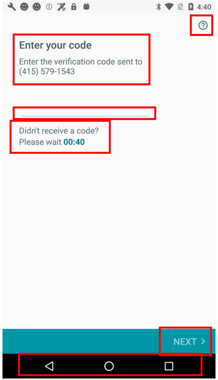
Proposed Methods: Grading & Optimizing

Proposed Solutions



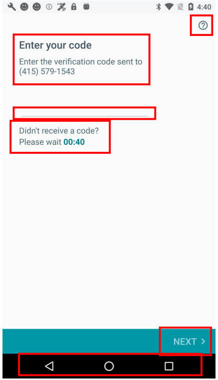
- Identify functional elements of User Interface

Proposed Solutions



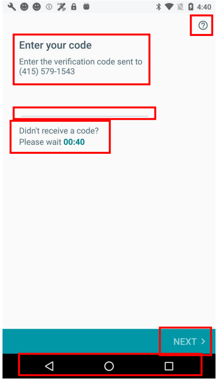
- Identify functional elements of User Interface
  - e.g. Buttons, Text, Figures

Proposed Solutions



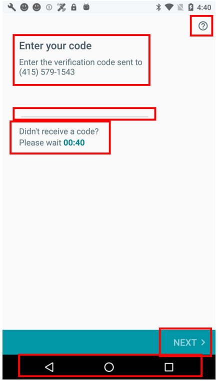
- Identify functional elements of User Interface
  - e.g. Buttons, Text, Figures
- Automated process comes up with missing pieces

Proposed Solutions



- Identify functional elements of User Interface
  - e.g. Buttons, Text, Figures
- Automated process comes up with missing pieces
  - Layout, Background color

Proposed Solutions



- Identify functional elements of User Interface
  - e.g. Buttons, Text, Figures
- Automated process comes up with missing pieces
  - Layout, Background color
- Automated Grading of UIs via pretrained model to alleviate difficulties of defining what is considered ”pretty”

Proposed Methods: Overview

1. General Idea & Datasets
2. Experiment 1: Finetuning Stable Diffusion
3. Experiment 2: Affine Transformation Matrix as latent space
4. Experiment 3: Variational Auto-Encoders

General Idea: Grading Aesthetics as a Regression Problem

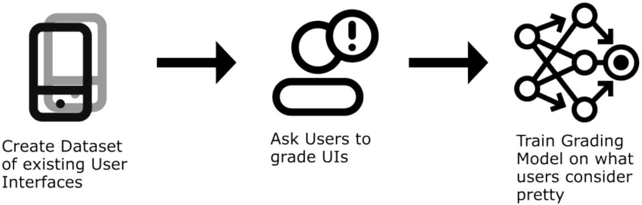


Figure 6: Grading mechanism

Dataset Collection

Biggest mobile UI Dataset: RICO

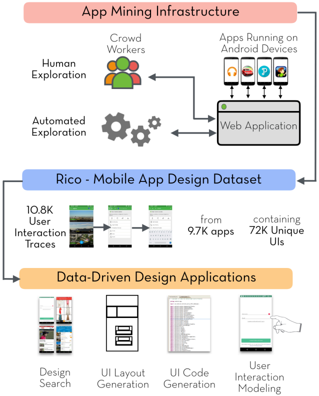


Figure 7: Rico Dataset: Automated Dataset collection

Dataset Collection

- Leveraging existing research by de Souza Lima et al. [2]
  - User study for grading on scale 1-5
  - Proposed model architecture: Finetuning Resnet-50
  - Only **2000** datapoints
  - Subset of the RICO dataset

General Idea: Optimizing

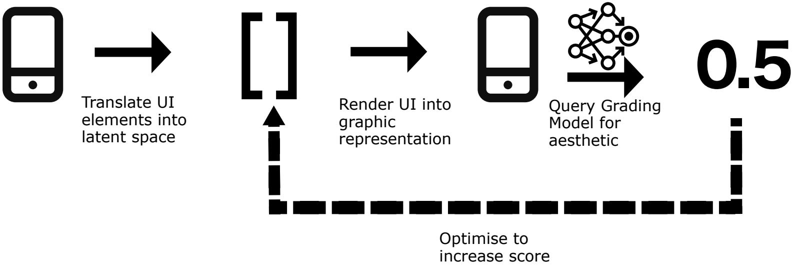


Figure 8: Optimizing mechanism

Experiment 1

- Operating directly on Pixel space: Fine-tuning Stable Diffusion

Experiment 1

- Operating directly on Pixel space: Fine-tuning Stable Diffusion
- Existing dataset of UI screenshots captioned with Salesforce BLIP

Experiment 1

- Operating directly on Pixel space: Fine-tuning Stable Diffusion
- Existing dataset of UI screenshots captioned with Salesforce BLIP
- Results not satisfiable

Experiment 1

- Operating directly on Pixel space: Fine-tuning Stable Diffusion
- Results not satisfiable

Figure 9: “UI” generated by SD model

Experiment 2: Element Positions as Latent Space

- Latent space: Position of UI elements

Experiment 2: Element Positions as Latent Space

- Latent space: Position of UI elements
- RICO dataset contains these information, translation is therefore straight forward

Experiment 2: Element Positions as Latent Space

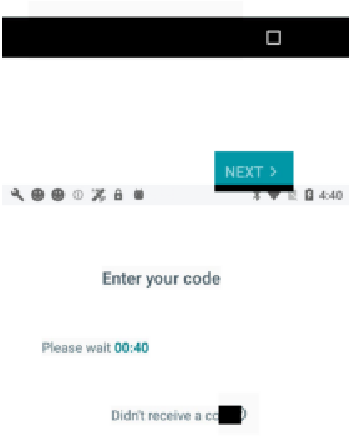
- Latent space: Position of UI elements
- RICO dataset contains these information, translation is therefore straight forward
- Practical setup: Vector containing positions of UI elements is considered a trainable parameter of a machine learning model

Experiment 2: Element Positions as Latent Space

- Latent space: Position of UI elements
- RICO dataset contains these information, translation is therefore straight forward
- Practical setup: Vector containing positions of UI elements is considered a trainable parameter of a machine learning model
- Assembly of final user interface and grading via model is done in a differentiable way
  $\rightarrow$ Task is classic machine learning problem

Experiment 2: Results

Start with random alignment:



Experiment 2: Results (ctd.)

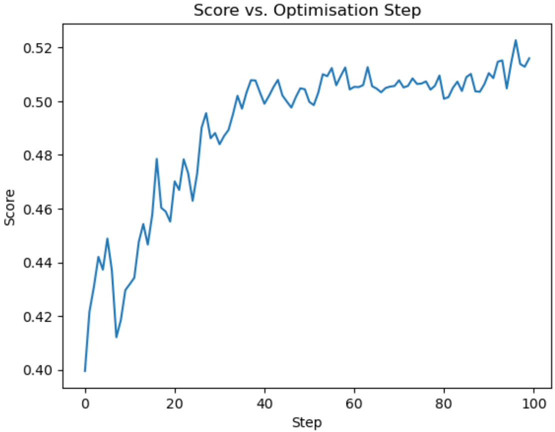


Figure 11: Score progression

Experiment 2: Results (ctd.)

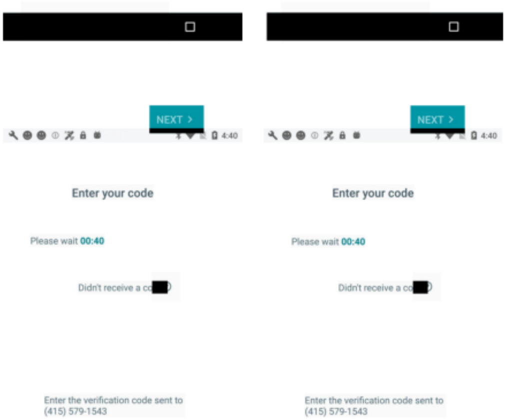


Figure 12: Before vs. After Optimization

Experiment 2: Results (ctd.)

- Hardly anything has changed

Experiment 2: Results (ctd.)

- Hardly anything has changed
- Model grades UI as aesthetic, when it is in fact “distorted”

Experiment 2: Results (ctd.)

- Hardly anything has changed
- Model grades UI as aesthetic, when it is in fact “distorted”
- Optimizer quickly learns weaknesses of grading model

Experiment 2: Results (ctd.)

- Hardly anything has changed
- Model grades UI as aesthetic, when it is in fact “distorted”
- Optimizer quickly learns weaknesses of grading model
- Second (potential) issue: Aesthetics classifier can’t distinguish between “real” and “fake” layouts

Experiment 2: Results (ctd.)

- Hardly anything has changed
- Model grades UI as aesthetic, when it is in fact “distorted”
- Optimizer quickly learns weaknesses of grading model
- Second (potential) issue: Aesthetics classifier can’t distinguish between “real” and “fake” layouts
- Functionality has similar issues

Experiment 2: Results (ctd.)

- Hardly anything has changed
- Model grades UI as aesthetic, when it is in fact “distorted”
- Optimizer quickly learns weaknesses of grading model
- Second (potential) issue: Aesthetics classifier can’t distinguish between “real” and “fake” layouts
- Functionality has similar issues

Going Forward

- Initial idea: Add additional classifier to detect “random” fake layouts
  $\rightarrow$ Not conclusive

Going Forward

- Initial idea: Add additional classifier to detect “random” fake layouts
  $\rightarrow$ Not conclusive
- One (other) approach to alleviate:
  $\rightarrow$ Reduce dimensions of or change characteristics of latent space

Experiment 3 (WIP)

- Automatically find a suitable latent space
  $\rightarrow$ Variational autoencoder (VAE)

Experiment 3 (WIP)

- Automatically find a suitable latent space
  $\rightarrow$ Variational autoencoder (VAE)
- Has the advantage of only producing valid “real” UIs

VAE for Enforcing Valid UI Generation

- General Idea:

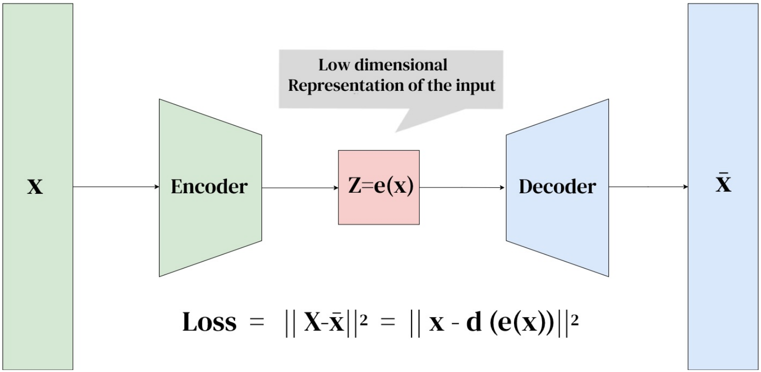


Figure 13: VAE Schematic reproduced from mlarchive.com [7]

VAE for Enforcing Valid UI Generatio

- In our case: $\mathbf{X}$ will be an $m \times 2$ vector containing

VAE for Enforcing Valid UI Generatio

- In our case: $\mathbf{X}$ will be an $m \times 2$ vector containing
  - first 5 clickable elements as relative positions on the canvas
  - first 5 non-clickable elements as relative positions on the canvas

VAE for Enforcing Valid UI Generatio

- In our case: $\mathbf{X}$ will be an $m \times 2$ vector containing
  - first 5 clickable elements as relative positions on the canvas
  - first 5 non-clickable elements as relative positions on the canvas
- ($m = 10$)
- 5 was chosen as sufficient dataset examples exist
- (m elements, each two coordinates)

VAE for Enforcing Valid UI Generatio

- In our case: $\mathbf{X}$ will be an $m \times 2$ vector containing
  - first 5 clickable elements as relative positions on the canvas
  - first 5 non-clickable elements as relative positions on the canvas
- $(m = 10)$
- 5 was chosen as sufficient dataset examples exist
- (m elements, each two coordinates)
- Optimization happens directly on latent space of the VAE

VAE for Enforcing Valid UI Generatio

- In our case: $\mathbf{X}$ will be an $m \times 2$ vector containing
  - first 5 clickable elements as relative positions on the canvas
  - first 5 non-clickable elements as relative positions on the canvas
- ($m = 10$)
- 5 was chosen as sufficient dataset examples exist
- (m elements, each two coordinates)
- Optimization happens directly on latent space of the VAE
- Second loss is potentially needed in order to keep the latent vector in the correct distribution

Outlook & Remaining Work During the Thesis

- Hardening aesthetics predictor against adversarial attacks

Outlook & Remaining Work During the Thesis

- Hardening aesthetics predictor against adversarial attacks
  - i.e. evaluating different model architectures

Outlook & Remaining Work During the Thesis

- Hardening aesthetics predictor against adversarial attacks
  - i.e. evaluating different model architectures
- Increasing aesthetics dataset size

Outlook & Remaining Work During the Thesis

- Hardening aesthetics predictor against adversarial attacks
  - i.e. evaluating different model architectures
- Increasing aesthetics dataset size

Other issues:

- “Phantom” elements in RICO dataset (potentially requires sanitization)

Future Work

- Optimization directly on code not only on arrangement
- Integration in production ready application
- Explore different latent spaces
- Optimize for different metrics
- Condition on usecase/functionality

Conditioning on Usecase

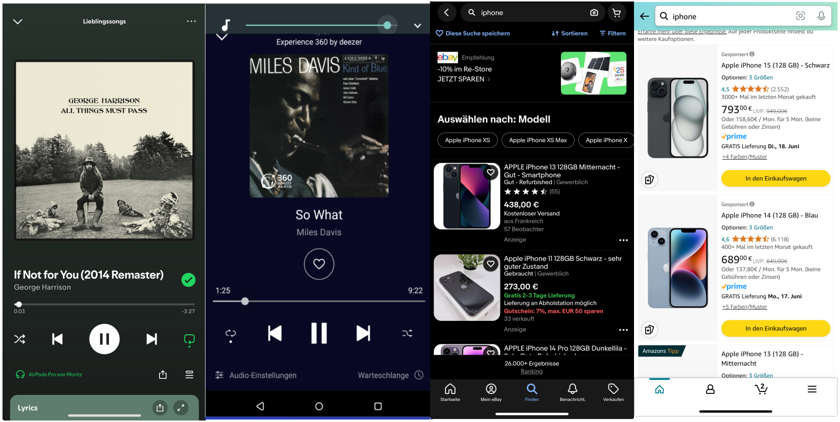


Figure 14: Similarities between apps of similar categories

Conclusion

- Objective: Optimize UI Layout to increase aesthetics

Conclusion

- Objective: Optimize UI Layout to increase aesthetics
- Experiments
  1. Stable Diffusion: Results not satisfactory
  2. Affine matrix as latent space: Too many degrees of freedom, results not satisfactory
  3. VAE: Ensure only “valid” UIs will be generated (In Progress)

Conclusion

- Objective: Optimize UI Layout to increase aesthetics
- Experiments
  1. Stable Diffusion: Results not satisfactory
  2. Affine matrix as latent space: Too many degrees of freedom, results not satisfactory
  3. VAE: Ensure only “valid” UIs will be generated (In Progress)

Thank you for your attention!

References I

[1] Maria Douneva, Rafael Jaron, and Meinald T. Thielsch. Effects of Different Website Designs on First Impressions, Aesthetic Judgements and Memory Performance after Short Presentation. *Interacting with Computers*, 28(4): 552–567, 06 2016. ISSN 0953-5438. doi: 10.1093/iwc/iwv033. URL `https://doi.org/10.1093/iwc/iwv033`.

[2] Adriano Luiz de Souza Lima, Osvaldo P Heiderscheidt Roberge Martins, Christiane Gresse von Wangenheim, Aldo von Wangenheim, Adriano Ferreti Borgatto, and Jean CR Hauck. Automated assessment of visual aesthetics of android user interfaces with deep learning. In *Proceedings of the 21st Brazilian Symposium on Human Factors in Computing Systems*, pages 1–11, 2022.

[3] Design Council. Design council, 2024. `https://www.designcouncil.org.uk/` [Accessed: June 2024].

References II

[4] Christiane Gresse von Wangenheim, João V. Araujo Porto, Jean C. R. Hauck, and Adriano Ferreti Borgatto. Do we agree on user interface aesthetics of android apps? *CoRR*, abs/1812.09049, 2018. URL `http://arxiv.org/abs/1812.09049`.

[5] Michael Bauerly and Yili Liu. Effects of symmetry and number of compositional elements on interface and design aesthetics. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50:304–308, 10 2006. doi: 10.1177/154193120605000320.

[6] Xiang Kong, Lu Jiang, Huiwen Chang, Han Zhang, Yuan Hao, Haifeng Gong, and Irfan Essa. BLT: bidirectional layout transformer for controllable layout generation. *CoRR*, abs/2112.05112, 2021. URL `https://arxiv.org/abs/2112.05112`.

References III

[7] mlarchive.com, 2024. https://mlarchive.com/deep-learning/variational-autoencoders-a-vanilla-implementation/ [Accessed: June 2024].

Additional Details

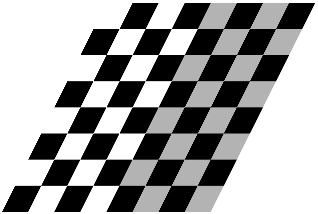
Experiment 2: Image translation

- Challenge: Differentiable Render

Experiment 2: Image translation

- Challenge: Differentiable Render
- Solution: Affine transformation:

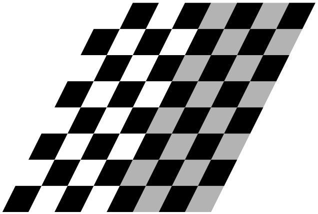
$$\begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Experiment 2: Image translation

- Challenge: Differentiable Render
- Solution: Affine transformation:

$$\begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$\rightarrow$ latent vector is affine matrix